1

Embedded device with software registry

5   The present invention relates to control devices of the kind that are embedded in electrical and electronic apparatus. The invention has been developed for application to mobile telephones but is equally applicable to a wide range of equipment with embedded devices from vehicles to washing machines.

10   Background

Software errors in a high volume commodity product (such as a mobile phone handset) can be very expensive to repair, involving a product recall, software reprogramming, and then re-issue of the product, not to mention such additional
15   items as replacements-under-guarantee and loss of reputation and customer confidence. Clearly, a mechanism is needed where software upgrades and fixes, known as "patches", can be installed, uninstalled and managed on a target device without recall of that target device and consequently with little impact to the user of the device. Additionally, the uses to which such a device can be put vary from
20   user to user, and it may be necessary to install, uninstall and manage new software modules in addition to those already resident on the device.

Once such a mechanism is in place, the patches and updates installed on the handset may be managed, both locally on the handset User Interface (if present) or
25   remotely over a communications link.

It is known for a device with large hard disc memory capacity, such as a personal computer, to have a program registry stored on the hard disc. This registry may include information relating to any programs or software upgrades recently
30   installed on the computer.

US-B-6434744 shows a system for a computer in which patching operations are performed on particular applications and a configurable database is updated with patching information. In a system of this type, the application of a patch is performed at run-time when applications and their patches previously stored on
5   hard disc are copied to volatile memory for use. Such an arrangement is not suitable for an embedded device such as that contained in a mobile telephone where power consumption has to be minimised and memory space conserved.

Hitherto, when a software modification is required on a mobile telephone (or other
10   device/appliance having an embedded system), for example during a product recall, it would be usual to rewrite the whole of the device software, for example by "reflashing" a flash memory. A separate management system might store information identifying which software versions have been installed on which devices, for example to avoid repetition and to aid future problem analysis.
15

The present invention provides a control device as defined in annexed claim 1. Preferred features of the device are described in the subsidiary claims. The installed programs may include equipment operating programs as well as universal applications such as calendars and calculators.
20

By contrast with the prior art computer system described above, in a control device such as for a mobile telephone, the operating programs are stored in and run from non-volatile memory rather than having to be transferred to volatile memory at run time. In order to conserve memory space, in a control device according to the
25   invention a new patch is used to modify the code to which it relates instead of being stored separately, and the registry simply contains a patch descriptor. There is then no need for the patch code itself to be separately stored.

An example of a patch registry suitable for use in a control device according to the
30   invention, sometimes called a "target device", will now be described by way of example only and with reference to the accompanying drawings in which:

Figure 1 is a block diagram over-view of a patch registry; and

Figure 2 is a more detailed block diagram showing an example of the data structures that may be used in the patch registry.

A device according to the present invention typically consists of a self-contained device having one or more computing components each with processor, memory and non-volatile program storage of limited size. These may not be "self contained" units and processors may share memory space. The device may also contain network connectivity to enable access to a server system; or facility to allow network connectivity to a server system, and may contain a removable storage device, for example an SD card.

The patch registry will contain information about patch files (not the patch code itself), installed patches and software updates. A patch file is the means of delivery of one or more patches to the device, and consists of a collection of data which contains information sufficient to allow a device to modify its program memory. A patch file may contain information about one or more patches, each of which may be composed of information about one or more changes to be made to the program memory of the target device. The patch file must be transferred to the target device before any modifications can occur. The patch file is used by a system (e.g. program) resident on the target device.

Some of the characteristics of the device impose particular requirements on the design of the patch registry. These characteristics are:

- Limited memory space both for persistent (i.e. that which remains present over a power off-on cycle) storage of program and configuration information, and also non-persistent (volatile) storage of other information.

- The main persistent storage medium, usually Flash memory, requires specialised processing in order to make modifications to its contents.

- The file system uses memory as its storage medium rather than being disk-based, and in a memory-constrained device this file system will consequently have limited storage capacity.

- The program of each computing element typically runs directly from the persistent storage of the device, unlike larger desktop computing devices that store their programs on a backup store (usually a hard disk) and then load the programs that are desired to be run into volatile (non-persistent) storage during their start up sequence.

- It may not be commercially acceptable that the mobile device be restarted as a consequence of the changes being applied.

The requirements imposed on the design of the patch registry by these characteristics are:

- The information stored describing the changes must be particularly detailed, though must at the same time be optimised to consume a minimal amount of storage.

- The Patch system must allow recovery from a power failure during an installation or uninstallation.

The mechanism proposed is extendable to be used for new software installation, for upgrade or bug fixing, or installation of new functionality.

The following describes how information relating to patches is installed in the program storage of a target device. The method of distribution and installation of a patch is not described here, but may use the network connectivity or the removable storage device.

Referring now to Figure 1, the patch registry will include at least status and progress information 1, to be described in more detail below, patch list 2 and a list of unused program storage blocks 3. The patch list 2 will usually contain a list of patch identifiers such as P1, P2 ... etc. Typically, for each patch, the registry will contain a record 20 of information relating to the patch including patch descriptor 21 and a list of changes being effected by the patch, a, b, c etc. For each change, a,

b, c etc there will be a change descriptor 23 containing further details of the change. For each unused program storage block n, n+1, n+2, n+3 there is a corresponding information block 31, 32, 33, 34 to be described in more detail below.

5

In general a patch consists of one or more individual changes to the program memory of a target device, replacing "faulty code" or "old code" with "repaired" code or "new code". Each of these changes is therefore made to non-volatile storage.

10

The patch system installs patches on the Target Device in two ways. For each change either the faulty program code is overwritten by the repair code, or the repair code is installed in an unused area of program memory and program flow is directed to this area when required and back to the main program as necessary. In 15 both cases, a record of information about the identity and location of the installed patch is created and maintained consisting of items P1 and 20 to 23 of Figure 1. A list of unused program storage for each processing element may also be manipulated items 3 and 31-34 of Figure 1.

20 From time to time it may be necessary to uninstall or remove patches from the Target Device. The previously described record of information 2 is used to uninstall the changes and to return any areas of program memory that were used back to the list of unused program storage 3. Once the unused areas have been returned the information about the patch is removed from the Registry.

25

Lastly, the Server System may interrogate the Target Device to determine what patches are installed and what capacity for further patches is available. This information may also be presented on demand through the user interface of the Target Device (not shown). The information used to respond to such requests is 30 derived from information saved in the Patch Registry.

In all these cases, it is necessary for the Target Device to retain information about the patches installed in it, and to maintain information about the remaining unused program memory. The patch registry is the means by which this information is retained.

5

An exemplary data structure of the patch registry is shown in more detail in Figure 2. Any suitable storage format such as text, binary, XML may be used and will not be described in detail herein.

10   Status and progress information block 1 includes two elements, namely element 10 indicating the overall status of the registry, e.g. a counter value implemented at each update, and patch installation status information block 11 containing information about the progress of the installation of any particular patch. Patch descriptor information block 21 contains a simple patch identifier (ID) as well as a

15   text (TXT) descriptor element for presentation to the user by the device man-machine interface. The list of modified code descriptor elements 22 will contain, as well as the simple list illustrated in Figure 4, indicated by "head" item 25 and "tail" item 26, a count of the number of elements 27. Each change descriptor 23 will include item 200 identifying the processing element to which the patch is to

20   be applied, item 201 containing the address of the faulty code to be repaired, item 202 containing the size of the faulty code to be repaired, item 203 containing the address of the repair and item 204 containing the size of the repair. The executable code of the patch, item 100 will be at the address indicated in item 203.

25   The unused program storage may include literally "empty" memory space, erased storage, storage that is simply deallocated and not erased in which case the contents are not useful, and any other space that has no anticipated code use. Each list of unused program memory blocks holds blocks in size order, smallest first, and may contain a reference or pointer to each unused program memory block in

30   the list.

For each unused program storage block 31 to 34, the data structure will include the block address and block size. Two of these are shown, namely block address 341 for block 34, block size 342 for block 34, block address 311 for block 31 and block size 312 for block 31. Each block B1, B2, B3 etc will be held in the list 350 5 including a header item 351. The block list will be an array associated with a particular processing element M1, M2, ... Mn identified in list 360.

It will be appreciated from the foregoing that the patch registry satisfies the following requirements:

10

Record details of all patches already installed for interrogation remotely by a Server System.

Record details of all patches already installed for interrogation locally by management software on the Target Device.

15   For each completed change, identify the computing component(s) to which the change was applied.

Allow identification of attempts to send duplicate patches perhaps by retaining an ID for each installed patch.

Allow identification of address clashes between any installed patch and a

20   new patch.

Hold information about the unused program storage remaining for each computing element, and allow interrogation of same to facilitate patch installation and removal and for interrogation by the Server System or management software on the Target Device. The unused program storage is

25   typically composed of a number of individual blocks. The Registry lists these in a manner which facilitates efficient searching when installing a new patch.

Record details of patches being installed or uninstalled, to allow recovery after a power failure.

30

Since the Patch Registry is contained in non-volatile storage, and contains details about the patches installed over a particular version of the software of the target device, it should be noted that the patch registry must be rendered to an empty state when a new version of software is installed on the target device.

5